Supporting Information:

3-D Inorganic Crystal Structure Generation and Property Prediction via Representation Learning

Callum J. Court[1,+], Batuhan Yildirim[1,+], Apoorv Jain[1,2], and Jacqueline M. Cole[1,2,3 *]

[1] *Cavendish Laboratory, Department of Physics, University of Cambridge, J. J. Thomson Avenue, Cambridge, CB3 0HE, UK*

[2] *Department of Chemical Engineering and Biotechnology, University of Cambridge, West Cambridge Site, Philippa Fawcett Drive, Cambridge, CB3 0FS, UK*

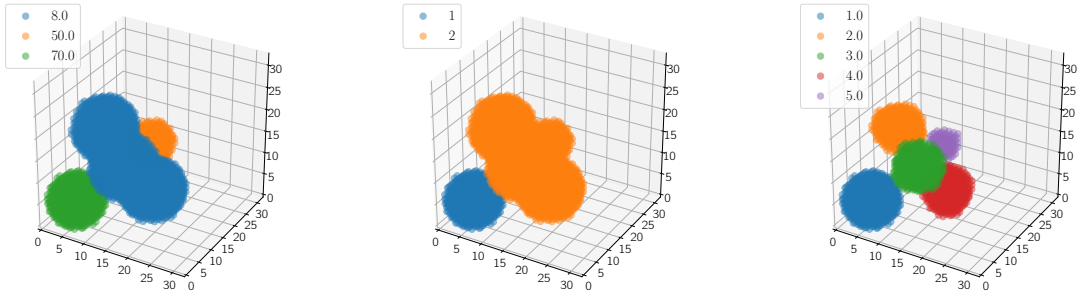
[3] *ISIS Neutron and Muon Source, STFC Rutherford Appleton Laboratory, Harwell Science and Innovation Campus, Didcot, OX11 0QX, UK*

[*] jmc61@cam.ac.uk

[+] Equal Contribution

July 30, 2020

1 Morphological Atom Segmentation



(a) (b) (c)

Figure S1: Overview of our morphological segmentation pipeline. (a) Example of a species matrix of a general perovskite $YbSnO_3$. (b) All connected components of the species matrix are first labeled. This yields regions that contain one or more clusters. (c) For each cluster, if the convexity is above 0.8, the class is determined to be a single cluster. Otherwise the region is iteratively eroded and segmented using the Watershed algorithm, in order to isolate individual clusters.

The UNet architecture yields a reconstructed species matrix, S', where each voxel therein is labeled according to the atomic number of the contained site (or zero otherwise). In addition, the UNet gives the corresponding binary mask of the segmentation, S'_B.

An example of a species matrix for a perovskite $YbSnO_3$ is shown in Figure S1(a). To convert this to atomic coordinates, the centroids of the labeled regions need to be determined. It is clear to a human that this matrix contains five atomic sites. However, automatic determination of the centroids is non-trivial owing to the overlap of atomic sites, by virtue of labeling all voxels within the ionic radius. In this work, we treat this problem as one of unsupervised clustering, whereby we attempt to find the number of clusters (atoms) and their centroid voxels to determine the final atomic coordinates. We have implemented a novel morphological clustering algorithm based on Watershed segmentation[1] taking ideas from Abdolhoseini *et al.*[2]. The basic idea of the algorithm is to find clusters that maximize the convexity of individual clusters.

The full algorithm is outlined in Algorithm S1. Starting with the binary mask of the

species matrix, all connected components are labeled in order to distinguish high-level cluster regions. Each of these sub-regions is cropped from the original image to determine the convexity according to

$$\text{convexity}(R) = \frac{\text{nonzero}(R)}{\text{nonzero}(R_{chull})} \tag{1}$$

where R is a region of the image, R_{chull} is the convex hull of said region and *nonzero* counts the non-zero voxels.

If the convexity is above a pre-defined threshold (taken as 0.8 herein), then the region is taken to be a single cluster and the process moves onto the next region. If instead the region is below the minimum convexity threshold, then it is deemed to contain multiple clusters and the Watershed algorithm attempts to segment them further. To do this, single dilation and erosion steps are executed on the region to form the certain background and foreground regions, with the difference being an "unknown" region that needs to be labeled. Watershed segmentation is carried out using the certain foreground as markers. This yields labeled sub-clusters that may or may not contain multiple atomic sites. The whole algorithm then repeats up to a defined number of iterations (five herein). The process also terminates if the size of the clusters becomes too small (less than eight voxels) as a result of the erosion steps.

After Watershed segmentation, regions corresponding to individual clusters were identified. The atomic sites were then determined by finding the centroid of each region. The corresponding atomic species is determined by majority voting, whereby the Watershed regions are compared to S and the most common voxel label is chosen as the atomic species.

We evaluated this method on the out-of-sample crystal structures, the results of which are displayed in Figure S2. This demonstrates that the method correctly estimates the number of atoms in the vast majority of cases.

For reference, prior to achieving this successful outcome using the Watershed algorithm,

Algorithm 1: Segment

Input:
species ▷ Species label matrix
binary ▷ Binary Species label matrix

Output: R

Params:
it ▷ Current Iteration
w_{min} ▷ Minimum object size
min_convexity ▷ Minimum cluster convexity

Segment
$labels$ = ConnectedComponents($binary$)
for *label in labels* **do**
  $bounding\_box$ = CalcBBox($label$)
  B_c = crop($binary$, $label$)
  S_c = crop($species$, $label$)
  $convexity$ = CalcConvexity(B_c)
  if $convexity \geq min\_convexity$: **then**
    $R[bounding\_box] = label$
    continue
  end
  else
    fg = erode(B_c)
    $markers$ = ConnectedComponents(fg)
    ws = WatershedSegmentation(B_c, $markers$)
    if $ws.size \geq min\_size$ *and* $it \leq max\_iters$ **then**
      ws = Segment(S_c, ws)
    end
    $R[bounding\_box] = ws$
  end
end
Return: R

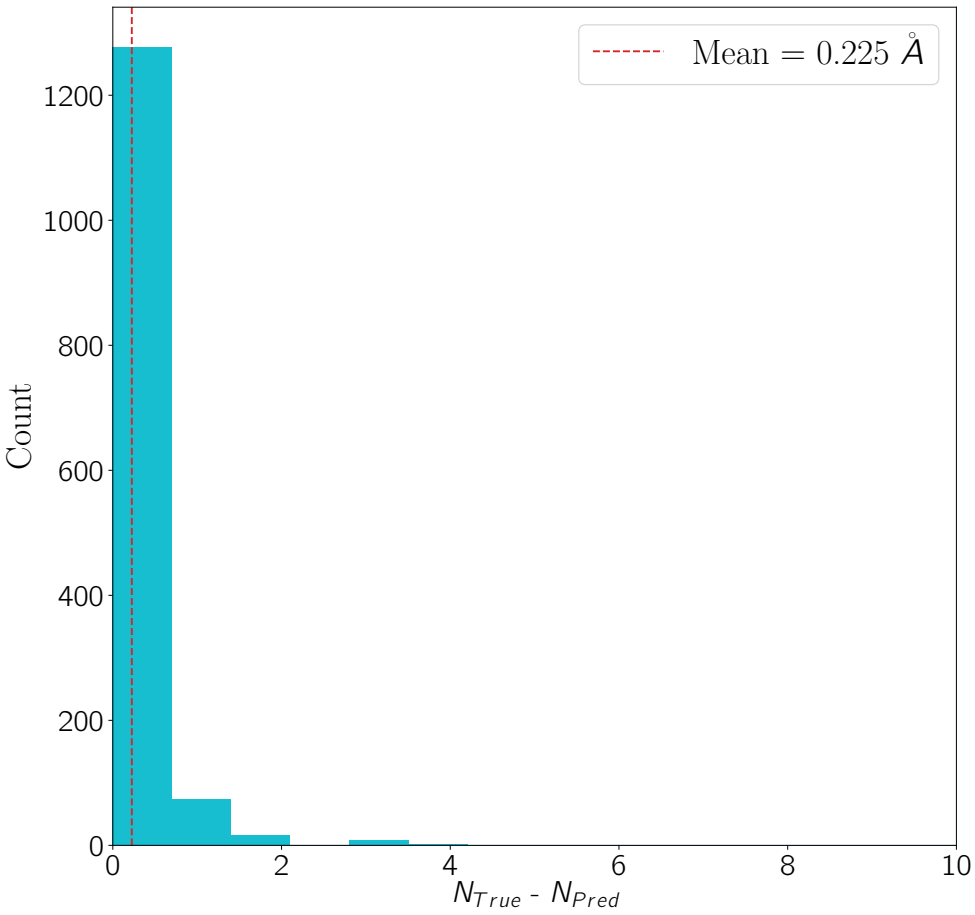


Figure S2: Out-of-sample validation of Watershed segmentation. Difference between the true and predicted number of atoms with our Watershed segmentation algorithm. The method correctly determined the number of atoms in nearly all cases.

we had explored various multiple off-the-shelf tools for unsupervised clustering including: DBSCAN[3] and K-Means[4]. However, these had little success. In most cases, the off-the-shelf tools under-estimated the number of atomic sites by failing to separate regions with high overlap. These failings further justify our choice of the Watershed algorithm.

2 Property-Prediction Implementation Details

2.1 Data

The MaterialsProject[5] database was used to train our property-prediction model, since it contains crystallographic information files (CIFs) and corresponding calculated properties for a large number of crystal structures. It is important to emphasize that these data include, but are not limited to, those used to train the VAE-UNet pipeline. In order to encourage reproducibility, we provide the MaterialsProject identifiers of the crystal structures that were used to train our property-prediction model, as well as a script used to obtain the crystal structures using the MaterialsProject API. These scripts are available, along with the full source code at `https://github.com/by256/icsg3d`.

2.2 Padding Crystal Graphs

Owing to the nature of the graph neural network (GNN) used for property prediction, we chose to pad all input crystal structures to sidestep the problem of variable-sized inputs to our GNN. We selected a maximum number of 50 atoms per unit cell for all crystal structures in the training set, and padded all crystal structures which had fewer atoms as follows. Firstly, for the node-feature matrix, we added several rows of zero vectors until the number of rows reaches 50. Since our node features are binary, this corresponds to adding several 'empty' nodes to the graph, in which none of the node features are present. Similarly for edge features, we pad the edge-feature tensors with zero vectors such that the resulting number of rows and columns is 50.

2.3 Masking Translations

It is necessary to mask translations during the graph transformations in order to ensure that nodes added by padding do not contribute to the learned representation and property predictions. For example, suppose that a crystal graph with several padded nodes is

passed through the GNN. When the node embeddings of the graph are projected by a weight matrix, no information is introduced to the padded nodes, since the matrix multiplication of a weight matrix with zero vectors still produces zero vectors. However, when a translation by a bias term follows this transformation, zero padded node-feature vectors have the potential to become non-zero. It is therefore crucial that these newly introduced spurious nodes are dealt with to ensure that the performance of the GNN is not hindered. To do so, we multiply the node-feature matrix by a binary-mask matrix, in which rows corresponding to nodes which exist in the graph have a value of 1, and rows which correspond to padded rows in the graph have a value of 0. After each translation in our GNN, we multiply our node-feature matrix by this mask, to ensure that the structure of the graphs input to the model are consistent throughout.

2.4 Pooling Padded Graphs

Average pooling is used in our GNN, owing to the padding present in our crystal graphs. Naively averaging over the nodes of the node-embedding matrix would produce an incorrect result, since zero-padded 'nodes' would contribute to the average over all nodes. Thus, we solve this problem by simply averaging over the true nodes in each graph, by summing over all nodes (padded nodes do not contribute to the sum) and dividing by the number of true nodes in each graph.

2.5 Crystal-Graph Features

The inputs to the GNN are computed directly from CIFs, and are as follows:

- An $N \times F_v$ node-feature matrix, whereby each row is an F_v-dimensional feature vector for each of the N atoms in the crystal.
- An $N \times N \times F_e$ edge-feature tensor, consisting of $N \times N$ edge-feature vectors of dimensions F_e.

- An $N \times E$ node-neighbor index matrix, denoting the indices of neighbors of each node in the node-feature matrix, where E is the predefined number of neighbors of each node in the graph.

As for the node and edge features that make up our inputs, we use the same features as Xie and Grossman[6]. The list of node features used is: group; period; electronegativity; covalent radius; number of valence electrons; first ionization energy; electron affinity; block (s, p, d, f); atomic volume. Edge features are simply discretized distances between pairs of atoms.

2.6 Training Procedure for Structure-Property Models

For properties where fewer training samples were available, we employed transfer learning and fine-tuning to improve the predictive capability and generalizability of these models. This includes the bandgap, bulk modulus, shear modulus, Poisson ratio, dielectric constant and refractive index models. For the pre-trained bases of these models, we utilised the weights learned by the formation energy model, which was trained on over 35000 samples. This includes the very first layer of the model, which is a fully-connected dense layer, and any following graph convolution layers. We then freeze the weights of these layers, and train to update the weights of the remaining fully-connected layers which follow the graph layers for 120 epochs. Following this, we unfreeze all layers (taking care to keep the parameters of the batch normalisation operations in the graph layers frozen), and fine-tune the entire model for 10 epochs with a learning rate that is decreased by a factor of 100.

3 List of Candidates

Table S1 shows the full list of candidate compounds and their associated predictions.

Table S1: Full list of candidate compounds generated using our pipeline alongside corresponding formation energy predictions and DFT validation results.

Formula	a Å	b Å	c Å	$E_{f,DFT}$ (eV/atom)	$E_{f,CGCNN}$ (eV/atom)	$\lvert\Delta E_f\rvert$ (eV/atom)	$\lvert\Delta Bonds\rvert$ (%)
$ZnPa$	4.94	4.94	5.15	0.42	0.38	0.04	0.00
KCa	4.42	4.19	4.2	0.55	0.60	0.05	0.00
$CoAs$	3.28	3.39	3.08	0.56	0.62	0.06	13.85
$SmCr$	4.05	3.98	4.27	0.47	0.53	0.06	1.71
KWO_3	4.05	4.06	3.85	-1.83	-1.90	0.07	3.26
$CaReO_3$	4.38	4.43	4.52	-1.36	-1.43	0.07	10.24
$AlSiO_3$	3.78	3.6	3.62	-1.99	-2.07	0.08	4.91
Rh_2NdHo	5.43	5.36	5.42	-0.31	-0.23	0.08	0.19
$ScWO_3$	4.61	4.74	4.55	-0.78	-0.88	0.10	11.44
YVO_3	4.02	4.18	4.18	-2.02	-2.16	0.14	1.58
$MgWO_3$	3.88	4.05	3.88	-1.80	-1.62	0.18	0.46
$PrMgO_3$	3.83	3.78	3.89	-2.64	-2.82	0.18	0.78
$LuPO_3$	4.15	4.19	4.06	-1.90	-1.71	0.19	12.82
Pd_2NdEu	5.18	5.23	5.12	-0.84	-0.63	0.21	0.21
$YReO_3$	4.4	4.02	4.36	-1.40	-1.62	0.22	9.39
$TiAlO_3$	3.86	3.88	3.95	-1.70	-1.94	0.24	3.08
YbV	3.6	3.72	3.97	0.76	0.50	0.26	3.19
$ErPO_3$	4.14	3.83	4.17	-2.20	-1.93	0.27	11.61
$CeBO_3$	3.38	3.36	3.58	-2.18	-2.46	0.28	6.98

Formula	a Å	b Å	c Å	$E_{f,DFT}$ (eV/atom)	$E_{f,CGCNN}$ (eV/atom)	$\lvert\Delta E_f\rvert$ (eV/atom)	$\lvert\Delta Bonds\rvert$ (%)
$CrAlO_3$	3.54	3.63	3.9	-1.80	-2.09	0.29	4.61
$LiWO_3$	4.41	3.94	4.08	-1.30	-1.60	0.30	11.34
MgN	3.02	3.54	3.46	0.42	0.02	0.40	23.05
$GdMgO_3$	3.75	3.66	3.82	-2.40	-2.84	0.44	0.53
$TiCrO_3$	3.87	3.64	3.75	-1.28	-1.84	0.56	0.53
$HfSiO_3$	3.95	3.72	3.95	-2.40	-1.84	0.56	8.26
$MgPO_3$	3.47	3.44	3.57	-1.81	-2.38	0.57	0.52
$TiSiO_3$	3.74	3.57	3.58	-1.44	-2.01	0.57	5.23
$ScReO_3$	4.51	4.28	4.25	-0.60	-1.23	0.63	11.27
$ScPO_3$	3.78	3.75	3.54	-1.51	-2.19	0.68	9.76
$TbMnO_3$	3.88	4.04	3.85	-1.59	-2.31	0.72	1.27
$TmMgO_3$	4.16	3.95	3.9	-1.84	-2.60	0.76	1.25
$LuMnO_3$	3.92	4.04	3.49	-1.28	-2.08	0.80	1.26
$LiReO_3$	3.92	3.76	3.67	-1.15	-1.98	0.83	0.69
$YIrO_3$	4.7	4.61	4.71	0.30	-0.77	1.07	12.20
Al_2MnOs	3.85	3.87	3.89	0.22	-0.86	1.08	1.81
Ru_2ScAl	4.21	4.19	4.26	0.39	-0.72	1.11	0.40
$SmCo$	3.8	3.99	3.27	-1.46	-0.32	1.14	4.61
$CaBeO_3$	3.73	3.89	3.82	-1.28	-2.46	1.18	2.62
$ErMnO_3$	4.26	4.02	3.91	-0.96	-2.14	1.18	10.83
$MoAs$	4.15	3.9	3.93	2.22	0.97	1.25	27.80
$TaSb$	3.92	3.88	3.88	1.94	0.67	1.27	13.36
$CeBe$	3.52	4.17	3.45	1.60	0.30	1.30	3.23
EuV	3.63	3.65	3.95	-1.08	0.50	1.58	3.21
$CeMgO_3$	3.59	3.57	3.26	-1.52	-3.11	1.59	4.03

Formula	$\boldsymbol{a}$ **Å**	$\boldsymbol{b}$ **Å**	$\boldsymbol{c}$ **Å**	$\boldsymbol{E_{f,DFT}}$ **(eV/atom)**	$\boldsymbol{E_{f,CGCNN}}$ **(eV/atom)**	$\boldsymbol{\lvert\Delta E_f\rvert}$ **(eV/atom)**	$\boldsymbol{\lvert\Delta Bonds\rvert}$ **(%)**
Pb_2PrCa	5.44	5.33	5.19	1.10	-0.53	1.63	2.35
$NaTl$	4.11	4.16	3.89	1.79	0.06	1.73	0.99
VI	3.76	3.38	3.88	2.39	0.56	1.83	7.35
CrI	4.2	4.13	4.38	2.32	0.43	1.89	23.60
$ScZnO_3$	3.89	4.02	3.82	0.01	-2.39	2.40	1.20
Sc_2MnAs	4.47	4.23	4.16	1.68	-0.76	2.44	0.70
$ZnIn$	5.00	4.94	4.94	3.54	0.68	2.86	17.74
Y_2ErAg	5.55	5.43	5.43	3.07	0.09	2.98	0.48
Ho_2EuPd	5.33	5.48	5.27	2.74	-0.27	3.01	0.54
Ho_2YbCd	5.09	5.1	4.98	2.82	-0.22	3.04	0.83
Pd_2CdEu	4.98	4.86	4.85	2.45	-0.73	3.18	0.92
Er_2HoAg	5.14	5.04	4.92	3.15	-0.20	3.35	0.40
Sc_2RuRh	4.09	4.22	4.36	2.59	-0.95	3.54	0.50
$ZnSb$	4.25	4.34	4.61	4.02	0.20	3.82	15.00
$LiCd$	3.82	3.87	3.28	3.89	0.06	3.83	5.74
$NdFe$	4.01	4.03	3.99	-3.39	0.49	3.88	1.25
Pd_2YbSm	5.26	5.21	5.17	3.28	-0.60	3.88	0.19
Pd_2EuTm	5.43	5.3	5.24	3.64	-0.43	4.07	0.94
$NiSb$	3.66	3.79	3.59	5.12	0.82	4.30	20.38
Dy_2ZnAu	5.31	5.11	5.32	4.71	-0.27	4.98	1.07
Ho_2ZnAu	5.26	5.24	5.12	4.79	-0.31	5.10	2.50
$InAu$	5.02	4.88	4.75	5.75	0.60	5.15	12.08
Pm_2ZrPd	5.14	5.03	5.11	-5.62	-0.32	5.30	10.80
$CuSb$	3.97	3.79	4.05	6.32	0.95	5.37	12.19
Lu_2ZnAu	5.16	5.27	5.17	5.25	-0.25	5.50	1.54

Formula	$\boldsymbol{a}$ **Å**	$\boldsymbol{b}$ **Å**	$\boldsymbol{c}$ **Å**	$\boldsymbol{E_{f,DFT}}$ **(eV/atom)**	$\boldsymbol{E_{f,CGCNN}}$ **(eV/atom)**	$\boldsymbol{\lvert\Delta E_f\rvert}$ **(eV/atom)**	$\boldsymbol{\lvert\Delta Bonds\rvert}$ **(%)**
$GaCu$	3.93	3.55	3.98	6.42	0.88	5.54	2.62
Au_2TmPb	5.63	5.39	5.74	6.09	-0.01	6.10	1.07
Pd_2LaAu	5.13	5.32	5.34	5.87	-0.28	6.15	0.30
$NiRh$	3.89	3.96	4.01	8.12	1.36	6.76	32.88
$ZnAu$	5.06	4.98	4.96	9.92	0.60	9.32	5.60
Ho_2AgAu	5.5	5.31	5.18	x	-0.58	x	x
Ho_2CdPd	5.16	5	4.81	x	-0.60	x	x
Average						**1.99**	**5.99**

4 Materials Project and ICSD Records for Polymorphs of VAE-generated Crystal Structures

Table S2 below shows comparisons between the VAE-generated candidates and experimentally/computationa determined polymorphs present in the ICSD and Materials Project database.

Table S2: Comparison of MAE between VAE-generated candidates and non-cubic polymorphs present in the MaterialsProject Database.

MAE Candidate	***a*** **Å**	***b*** **Å**	***c*** **Å**	E_f **eV/atom**	***ICSD ID***	***Crystal System***
$NiSb$	0.18	0.05	1.44	0.39	646419	hexagonal
$CoAs$	0.17	0.06	1.86	0.41	43888	hexagonal
$LuMnO_3$	2.05	1.94	7.09	0.59	280779	hexagonal
VI	0.31	0.69	4.45	0.82		hexagonal
$MgWO_3$	1.30	1.25	3.89	0.42		orthorhombic
$MoAs$	0.90	2.02	2.39	0.35	43188	orthorhombic
$TiCrO_3$	1.13	1.63	3.75	0.30		orthorhombic
$AlCrO_3$	1.37	1.30	3.50	0.19		orthorhombic
$ScAlRu_2$	4.98	6.44	6.36	3.38		orthorhombic
$ErMnO_3$	0.89	1.69	3.41	0.52	183141	orthorhombic
$TbMnO_3$	1.35	1.72	3.55	0.51	252442	orthorhombic
$ZnSb$	1.89	3.34	3.47	0.09	601137	orthorhombic
$CeBO_3$	1.62	2.36	4.53	0.51	99689	orthorhombic
YVO_3	0.16	0.45	0.44	0.34		tetragonal
$LiReO_3$	1.37	1.54	1.64	0.12	35012	trigonal
$NiRh$	1.37	1.44	0.31	0.06		trigonal
Average MAE	**1.48**	**1.99**	**3.48**	**0.59**		

5 Optimization of Delocalized Compounds

All machine-learning models are inherently biased by their training data. For this work, this means that our generative and predictive models are restricted by the limitations of the DFT calculations performed by the Materials Project. Such calculations often struggle to optimize in cases of compounds containing highly delocalized electrons, as present in certain rare-earth metals, the chalcogens, and the metalloids. Table S3 below shows the results of optimizing three such generated materials, $NaTe$, $TiSe$ and $AgTe$. As shown, in these cases, the bond lengths vary substantially from the predicted values, by over 38% in $TiSe$.

Table S3: Results of the geometry-optimizatied DFT calculations on compounds containing delocalized electrons.

Candidate	$\|\boldsymbol{\Delta_E}\|$ **(eV/atom)**	$\|\boldsymbol{\Delta_{bonds}}\|$ **(%)**	$\|\boldsymbol{\Delta_{cell}}\|$ **(%)**
$TiSe$	2.89	38.6	0
$NaTe$	1.21	26.7	0
$AgTe$	4.44	30.0	0

References

[1] Digabel, H.; Lantuéjoul, C. Iterative Algorithms. Proc. 2nd European Symp. Quantitative Analysis of Microstructures in Material Science, Biology and Medicine. 1978; p 8.

[2] Abdolhoseini, M.; Kluge, M. G.; Walker, F. R.; Johnson, S. J. Segmentation of Heavily Clustered Nuclei from Histopathological Images. *Sci. Rep.* **2019**, *9*, 1–13.

[3] Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Kdd. 1996; pp 226–231.

[4] Lloyd, S. Least Squares Quantization in PCM. *IEEE transactions on information theory* **1982**, *28*, 129–137.

[5] Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G. Commentary: The Materials Project: A Materials Genome Approach to Accelerating Materials Innovation. *Apl Materials* **2013**, *1*, 011002.

[6] Xie, T.; Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.* **2018**, *120*.